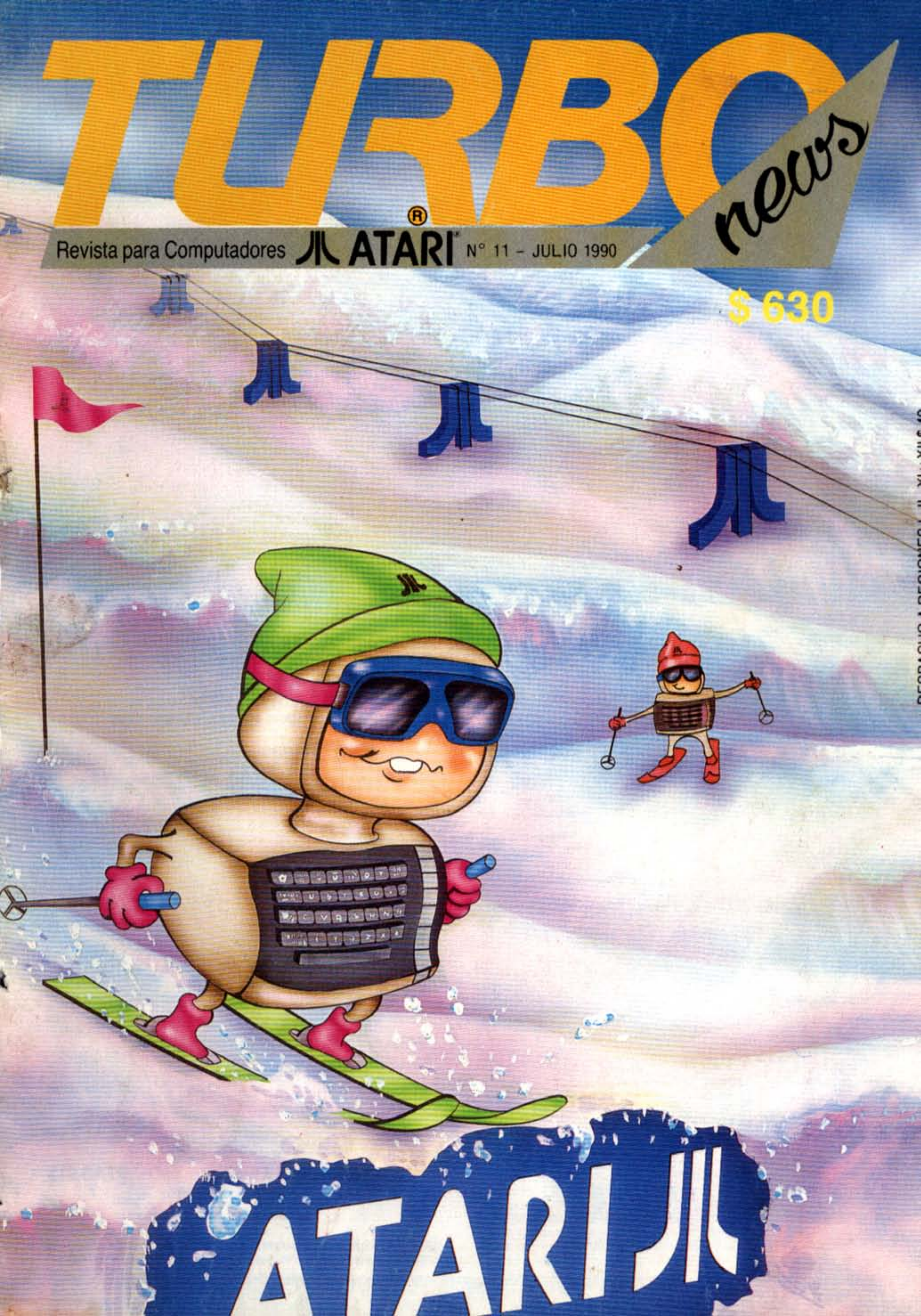
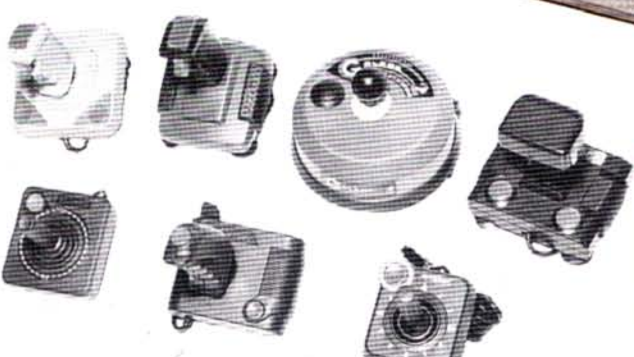

TURBO
news
Revista para Computadores ATARI N° 11 - JULIO 1990
$ 630
DESPACHO A REGIONES I, II, XI, XII $ 40
ATARI

COMPUTACION



1 COMPUTADOR 65 XE
1 CASSETTERA
2 JOYSTICKS
6 JUEGOS

$ 67.500.



JOYSTICK

JOYSTICK GALAGA	$ 3.750
BG - 201	
CON MICROSWITCH	$ 6.040
BG - 747	$ 3.000
BG - 105	$ 2.500
BG - 124	$ 1.500
QUICK SHOT I	$ 2.200
QUICK SHOT II (AUTODISPARO)	$ 5.250

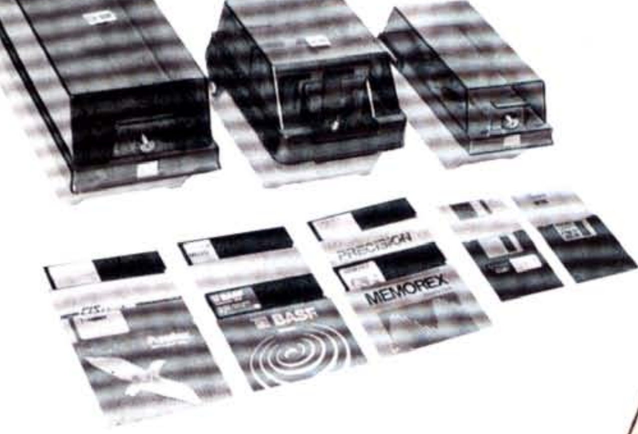
PROGRAMAS EDUCATIVOS

SERIE TM (TELEMATICA)
DESDE $ 1.790

CASSETTES EDUCATIVOS
TURBO SOFTWARE DESDE $ 895

CAJAS PORTADISKETTES

3.1/2 40 UNIDADES	$ 3.750
5.1/4 50 UNIDADES	$ 3.650
5.1/4 100 UNIDADES	$ 4.500



DISKETTES 5 1/5 2S-2D:

VERBATIM	$ 390 c/u.
FUJI	$ 355 c/u.
GOLDSTAR	$ 225
PRECISION	$ 225
MEMOREX	$ 310 c/u.

DISKETTES 3,5"

FUJI	$ 695
MEMOREX	$ 780
PRECISION	$ 715





PRIMER CENTRO ELECTRONICO CHILENO

AV. L. B. O'HIGGINS 845 **MONJITAS 813**
CENTRAL: 6322313

FAX: 399047 - TELEX 340517

DESPACHOS A PROVINCIA PREVIO ENVIO DE CHEQUE, VALE VISTA O GIRO TELEGRAFICO A CORREO 21 - CASILLA 395 -V -STGO.

•ELECTRONICA •ELECTRICIDAD •COMPUTACION •AUDIO - VIDEO

EDITORIAL

Queridos amigos:

En este número queremos compartir con Uds. una gran alegría. Debido a su continuo apoyo y afecto, hemos podido aumentar el número de ejemplares de nuestra revista lanzados al mercado. Esto nos permite, sobre todo, abarcar mejor las provincias, donde existe un gigantesco grupo de Atarianos que han estado siempre un poco aislados de todas las novedades que van surgiendo aquí en Santiago.

Hemos recibido noticias de grupos de fanáticos que, sin ningún apoyo y con sólo ese gran amor que sentimos todos por el Atari, se reúnen en los lugares más recónditos del país para intercambiar novedades. A todos ellos los invitamos a participar en nuestra Revista, para adquirir mayor difusión y aumentar su número de participantes. Escriban sin dudar, que sus cartas serán publicadas y sus clubes dados a conocer.

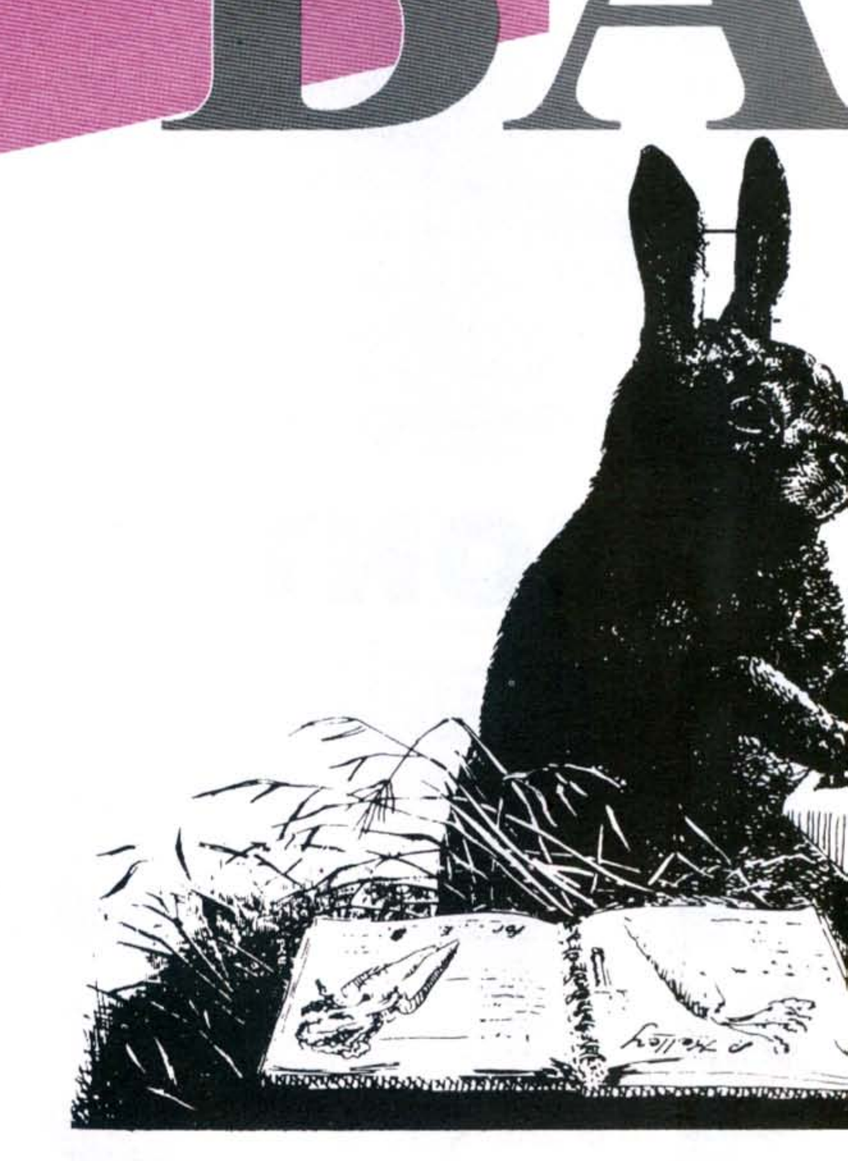
Pero basta de preliminares, ingresemos de lleno a las páginas de este nuevo número de tu Revista Turbo news.

CONTENIDO

TURBO® news

Circulación Mensual, Nacional e Internacional.
Destinada a los usuarios de computadores ATARI (R) como material didáctico de Programación. TURBO news (R) es una publicación de EDITORA TURBO LTDA. Domicilio: Av. Fco. Bilbao 4226 - Teléfono: 486506.

DIRECTOR RESPONSABLE: Mauro Pieressa. REPRESENTANTE LEGAL: Marcelo Waldbaum. PRODUCCION: Marcelo Waldbaum y Mauro Pieressa, Programadores y Diseñadores de Computación. DIRECTORA DE ARTE: Odalí Guerrero L. CORRECTOR: Marcial Valenzuela S. PUBLICIDAD Y RR.PP.: Liliana Muñoz Otárola, Hernán Vittini. COLABORACION: Mariana Pizarro. PUZZLE: Mario Calvo A. FOTOCOMPOSICION: Brubytes. IMPRESION: Servigraf. DISTRIBUCION: Alfa Ltda. Agradecemos la colaboración de COELSA S.A. Centro Atari. (Augusto Leguía Sur 75). Atari es marca registrada de ATARI CORPORATION. TURBO news es marca registrada de EDITORA TURBO LIMITADA. (Registro de Marca N° 342428 9-05-89).



En este número veremos los operadores lógicos utilizados por el lenguaje Basic.

Los operadores lógicos en programación, tienen el mismo uso que en el lenguaje cotidiano. Se utilizan para establecer comparaciones entre dos o más elementos. En los programas de los números anteriores los hemos utilizado instintivamente, ya que su comprensión es muy sencilla. En este artículo los detallaremos para que queden perfectamente determinados.

Primero veremos los operadores relacionales. Estos son:

SIMBOLO	EXPLICACION	EJEMPLO
<	MENOR	A<B
>	MAYOR	A>B
=	IGUAL	A=B
<=	MENOR O IGUAL	A<=B
>=	MAYOR O IGUAL	A>=B
< >	DISTINTO	A<>B

En caso de que la relación que se produzca sea verdadera, el computador le asigna un 1. Caso contrario, un 0. Esto puede ser utilizado perfectamente por el programador para reemplazar instrucciones IF. ¿Veamos cómo? Supongamos que tenemos la siguiente pregunta:

```
10 IF A>B THEN 120
20 PRINT "A NO ES MAYOR A B"
```

Podemos reemplazarla por lo siguiente:

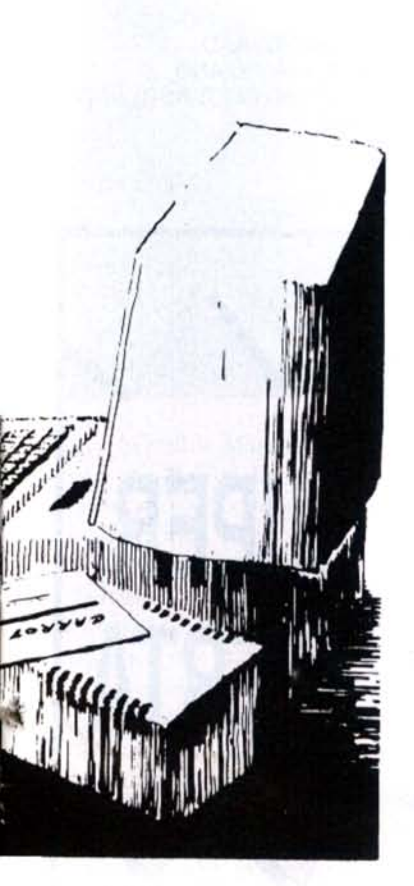
```
       (1)       (2)
10 GOTO (A>B)*120+(A<=B)*20
20 PRINT "A NO ES MAYOR A B"
```

Como sólo puede ser verdadero (1) o (2), pero no los dos al mismo tiempo, uno va a valer uno y el otro cero. Si A es mayor a B, el computador va a hacer:

```
10 GOTO 1*120+0*20
```

SIC

LECCION 11



o sea

10 GOTO 120

Que es lo que queríamos. Si A es menor o igual a B, va a quedar

10 GOTO 0*120+1*20

o sea

10 GOTO 20

Lo cual también es correcto. En este caso quizás no se vea ningún beneficio, pero hay otros donde sí. Por ejemplo:

```
10 IF I=7 THEN 100
20 IF I=9 THEN 200
30 IF I=13 THEN 300
```

En este caso no se puede utilizar la instrucción ON I GOTO, porque los valores de I no cumplen ninguna propiedad que se pueda expresar mediante una fórmula. En cambio sí podemos hacer:

```
10 GOTO
(I=7)*100+(I=9)*200+(I=13)*300
```

Ganando el programa en velocidad y memoria utilizada. Finalmente, tenemos los operadores lógicos. Estos permiten unir combinaciones de operaciones relacionales. Ellos son:

NOMENCLATURA	EXPLICACION	EJEMPLO
NOT	NO	NOT(A=B)
AND	Y	(A>B) AND (B>C)
OR	O	(A>B) OR (A>C)

Veamos ejemplos de sus utilizaciones:

```
IF NOT(A=B) THEN 100
```

Es lo mismo que haber hecho

```
IF A<>B THEN 100
```

```
IF (A>B) AND (B>C) THEN 100
```

Significa que si A>B y B>C, se va a ejecutar la instrucción 100

```
IF (A>B) OR (A>C) THEN 100
```

Significa que si A>B o mayor que C entonces se va a ejecutar la instrucción 100.

Estas instrucciones lógicas, también pueden utilizarse en instrucciones GOTO. Veamos qué valores asigna el computador con los siguientes ejemplos:

(A>B) AND	(A>C)	RESULTADO
0	0	0
1	0	0
0	1	0
1	1	1



BASIC

Para ver ejemplos de todo esto, analizaremos dos pequeñas rutinas que podrás incorporar a tus programas. Estas sirven para desplegar un menú de opciones, de manera mucho más "artística". Las dos, son pequeñas variaciones de lo mismo.

También en dichos programas utilizamos instrucciones POKE. Cualquier duda que surja, podrás consultar los artículos de mapa de memoria. La posición 710 se utiliza para indicarle al computador cuál va a ser el color de fondo, en este caso un 1 que significa negro con letras blancas.

La posición 752 se utiliza para hacer invisible al cursor. Para ver qué significa esto, ejecuta el programa quitando la instrucción POKE 752,1.

Finalmente para poder imprimir las flechas hacia arriba y hacia abajo en pantalla, debemos imprimir el caracter ESC (CHR$(27)) y luego la flecha hacia arriba o abajo (CHR$(28) O CHR$(29) respectivamente).

Recordando que (A>B) daba 1 si A era mayor que B y 0 en el caso contrario.

(A>B) OR	(A>C)	RESULTADO
0	0	0
1	0	1
0	1	1
1	1	1

NOT (A=B)	RESULTADO
0	1
1	0

Dentro de un programa quedaría:

```
10 IF (A>B) AND (A>C) THEN 100
20 IF (A>B) AND NOT(A>C)
THEN 200
30 IF NOT(A>B) AND (A>C)
THEN 300
```

NOTA: Hacer NOT(A>C) es lo mismo que hacer (A<=C)

Estas tres opciones pueden convertirse a:

```
10 GOTO ((A>B) AND
(A>C))*100+((A>B) AND
NOT(A>C))*200+(NOT(A>B) AND
(A>C))*300
```

ENTREVISTAMOS A:

INFORNA

"los niños son nuestro incentivo"



En este número hemos entrevistado a don Rafael Manzur, Gerente General de INFORNA S.A., empresa que desde hace 13 años está dedicada al rubro de Suministros de Computación, área en la cual son líderes.

El señor Manzur nos cuenta — en el área de los suministros, con nuestras ubicaciones en Av. Pedro de Valdivia entre Providencia y Costanera y en el Centro Galería Crillón, tenemos muchos clientes que tienen niños y ello nos ha obligado, por decirlo así, a entrar en el mundo de los Atari, pues es la puerta por donde la mayoría de los pequeños entran al fascinante mundo de la computación. Por esto es que para ellos tenemos equipos, cassetteras, disketteras y por sobre todo los excelentes programas educativos Turbo, tanto como sus juegos, en el denominado sistema nueva generación de Software para computadores Atari".

"Nuestro rubro que, como dije antes, es el suministro de computación de alta seguridad, comprende todos los diferentes productos necesarios para almacenar, leer e imprimir las







ENTREVISTAMOS A

INFORNA

"los niños son nuestro incentivo"



informaciones que proporcione el computador. Para ello contamos con la representación exclusiva para Chile de marcas de extraordinario prestigio, como Graham Magnetics, subsidiaria de Carlisle Memory Products Group en cintas magnéticas de muy alta seguridad; también representamos a D.E.I., de USA, en Data Cartridges, empresa que es la única licenciada de 3M para fabricar este producto y varias otras en Diskettes y accesorios. También somos distribuidores autorizados de IBM, en diskettes y data cartridges, de 3M."

"Aun cuando el mercado nacional se mueve a través de Santiago, en estos momentos hay provincias que son de sumo interés para nosotros por la envergadura que ha tenido su crecimiento y es por esto que próximamente estaremos en Antofagasta y Viña del Mar, desde donde esperamos atender a toda la Región del Norte y Quinta Región para seguir desarrollándonos en la atención de la mayor parte de nuestro territorio, con la misma calidad de servicio y eficiencia en nuestras entregas, ya que ellas son algunas de las cualidades de "valor agregado" que queremos brindar a nuestros numerosos y selectos clientes."

"A diferencia de otras empresas del rubro, nuestro interés es mantener una sólida especialidad en este rubro, pues creemos que es muy importante. Casi nadie piensa que el computador, sin un medio magnético eficiente, no sirve de nada, pues es en el medio magnético donde se graba y lee la información. He aquí el por qué de su importancia vital que casi nunca es reconocida y la razón primordial de que nuestra política de comercialización no varíe. En lo único que hemos transado un poco, para graficar mejor lo que queremos explicar, es en tener y mantener la línea Atari, por respeto a nuestros clientes que lo necesitan para sus hijos y porque pensamos que estamos dando un servicio eficiente a estos pequeños, vamos a tenerlos también como clientes cuando crezcan y se conviertan en los hombres de empresa del año 2000.

Apre
co

Matemática Espacial

El objetivo de este programa es que los niños practiquen las cuatro operaciones matemáticas básicas: suma, resta, multiplicación y división de números de hasta dos dígitos, en forma de un entretenido juego.

La nave madre está siendo atacada por feroces enemigos que intentarán destruirla, mediante el uso de misiles termonucleares. Tú cuentas con láseres defensivos que serán activados únicamente en el caso de responder correctamente a una operación matemática desplegada en pantalla. El tiempo de respuesta es esencial, ya que el misil no dejará de acercarse a tu nave a menos que respondas correctamente. Cada respuesta incorrecta hará que el misil se acerque más rápidamente.

El programa cuenta con varios niveles, los cuales están determinados por la velocidad de acercamiento del misil enemigo. La conducción del programa es muy simple, pues está preparada en la modalidad de menú.

ndiendo con tu mputador

A partir de este número, les comenzaremos a entregar la descripción de los programas educativos de la línea Turbo Software, que se encuentran en el mercado.

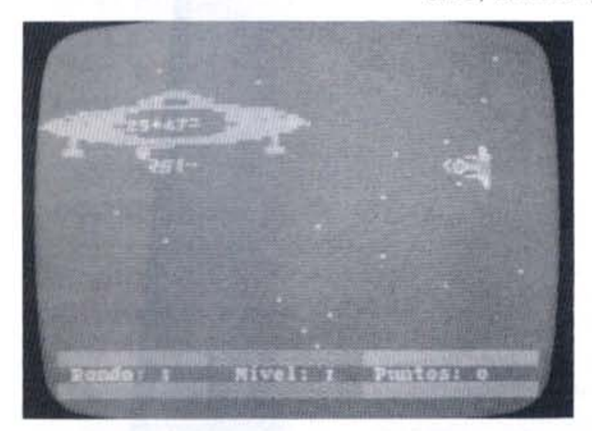
El oído y el sentido de la audición

El objetivo de este programa es que comprendas el funcionamiento del Oído junto con el sentido de la Audición. Puede acompañar al alumno de la enseñanza media a reforzar sus estudios de Biología.

El oído es una estructura muy compleja que no sólo permite percibir los sonidos, sino, también, las variaciones de movimiento y presión que experimenta nuestro cuerpo, todo ello demostrado en el programa.

Al finalizar el educativo, se te presentará un cuestionario con alternativas, con el objeto de evaluar la comprensión de la materia pasada.

Puedes utilizarlo en tu hogar, sin ayuda, puesto que el computador te irá guiando y corrigiendo hasta verificar la total comprensión de la materia estudiada.





TURBO mail

Después de un breve intervalo, reanudamos el contacto con nuestros lectores a través de esta sección de tu Revista Turbo news.

Sin lugar a dudas creemos que esta sección es muy importante pues permanentemente nos llegan a la Editorial numerosas cartas con colaboraciones y consultas de nuestros lectores. El problema que tuvimos en las ediciones anteriores fue puramente de espacio. Los cursos de Basic y Assembler van tomando cada vez más colorido y requieren de más páginas. Pero vamos manos a la obra para contestar las últimas cartas de nuestros aficionados Atarianos.

Sres. Turbo news:

Aprovecho la presente para felicitarlos por la excelente revista que están elaborando mes a mes. Soy una amiga que recién comienza a familiarizarse con la Programación en Lenguaje Basic, más explícitamente con el manejo de un computador Atari.

Su revista y los cursos que en ella se desarrollan me han motivado para hacer pequeños programas tales como, sacar un promedio, calcular una función entregando la Pre-imagen y el último que hice da la solución a ecuaciones cuadráticas, incluyendo raíces complejas que muchas calculadoras no las dan.

Me gustaría que en su prestigiosa revista publicaran un listado de mi programa, que podría ser de utilidad para amigos de enseñanza media y motivaría a otros, que también recién comienzan a programar, a modificarlo.

Me despido muy cordialmente y espero den buena acogida a mi solicitud.

Saluda atte. a Uds.

Patricia Barrera Lepe.

Patricia:

Gracias por tus palabras de aliento hacia nuestra: "La Revista Turbo news". Colaboraciones como la tuya son las que necesitamos del resto de la comunidad Atariana para seguir desempeñando nuestra tarea de unir a los Usuarios de nuestro amigo "El Atari".

Sin duda creemos que tu programa va a motivar a nuestros lectores a ingresarlo en sus computadores y a utilizarlo. Seguramente recibiremos más programas del estilo que también los publicaremos para que tú puedas analizarlos.

Con respecto a tu listado, a primera impresión parece estar muy bien estructurado. El analiza todos los casos posibles dentro de los resultados de las raíces cuadráticas de una ecuación. Pero tu programa puede ser modificado para que corra con menos instrucciones que las que tiene, así que te recomendamos que sigas estudiando el Lenguaje Basic para ir adquiriendo mayor dominio en la programación.

Marcelo Waldbaum

TECNICAS DE PROTECCION



En el número anterior, habíamos visto cómo evitar las copias realizadas por el DOS 2.5, pero también vimos que cualquier copiador de sectores podía eludir la protección. En este número, veremos cómo evitar también este tipo de copia.

Escondiendo el directorio de los discos, habíamos visto que no se podía saber cuál era el contenido de éste, ni tomar cualquiera de sus archivos en particular. Sin embargo, si quisiéramos tener el contenido de un disco así protegido, podríamos independizarnos de qué archivos posee y copiar su



TECNICAS DE PROTECCION

contenido tal cual a través de un copiador de sectores. No sabríamos dónde está cada uno de sus archivos, ni cómo se llaman, ni cuánto ocupan, pero tendríamos su copia y lo que es más importante, lo podríamos utilizar.

Una forma de diferenciar un original de una copia es creando los BAD SECTORS o sectores malos y hacer que el programa chequee que existan dichos sectores malos. Si no existen, significa que es una copia y por lo tanto no debería funcionar. Caso contrario, el programa funciona correctamente.

Un sector malo es básicamente un sector que no puede ser leído por la diskettera, arrojando un código de error. Puede producirse porque no está formateado, porque fue escrito por un cabezal desalineado, porque fue grabado a una velocidad que no correspondía o por falla física o de magnetismo. Cabe destacar que no se considera un sector malo, aquel que tiene información al azar o errónea, ya que la diskettera no se ocupa del contenido lógico de cada sector. Ese es problema del usuario.

Veamos cada uno de ellos.

Sector no normateado o mal formateado.

Lamentablemente la diskettera Atari no posee un comando que permita formatear un sector en particular. Cuando se le da una instrucción de formateo, formatea el disco entero. Una primera forma de hacer un sector no formateado es darle la instrucción de formateo y apagar la diskettera durante el proceso. Este método no permite controlar cuáles sectores van a quedar con error, pero puede utilizarse y después con un lector de sectores, averiguar cuáles quedaron malos. Otra manera de efectuarlo es utilizar un segundo computador que sí permita el formateo de algún sector en particular (un IBM PC, por ejemplo). El problema con este método es que como no coinciden los sistemas de formateo, tampoco se puede seleccionar un sector particular del disco Atari (aunque sí se puede hacer que caiga siempre en el mismo lugar), además de exigir el conocimiento del funcionamiento de un segundo computador.

Cabezal desalineado.

Este sistema, aunque funciona, no es recomendable puesto que una vez desalineado el cabezal, sólo puede volverse a alinear con el uso de un osciloscopio.

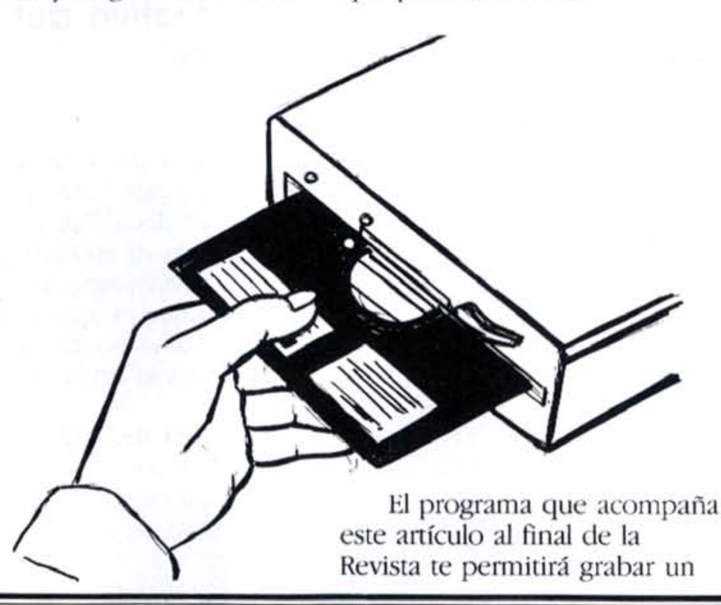
Daños físicos al disco.

Este método consiste en hacerse un mapa del diskette y su contenido y luego, con algún objeto punzante, dañar físicamente el disco en la posición deseada. Claro que dañar un sector exacto es sumamente difícil; además la protección daña el disco en forma permanente. Una forma de ocupar este sistema de protección, sin dañar el disco en forma permanente, es utilizar un magneto, siendo las probabilidades de dañar un sector en particular más pequeñas.

Reducir la velocidad de la diskettera.

Este método consiste en bajar la velocidad de la diskettera y luego darle instrucción al computador de grabar los sectores que deseamos arruinar. Existen programas con los que se pueden determinar la velocidad de rotación de la diskettera (el DOS 2.6, por ejemplo), siendo la velocidad standard 288 revoluciones por minuto. Girando el tornillo que se encuentra en la parte posterior del mecanismo se puede reducir la velocidad hasta 220 RPM, grabar los sectores deseados y luego volverla a la velocidad normal. Este método es sencillo ya que no cuesta mucho volver a ajustar la diskettera a la velocidad normal.

Otra manera más sencilla aún para crear un sector malo, es darle la instrucción de grabar un sector en particular y luego, en el momento que lo está realizando, tomar el disco por el extremo, como lo indica la figura y tirar repetidamente de éste hacia adelante y hacia atrás hasta que quede realmente malo. Las primeras veces que lo intentes, es muy probable que el sector no se dañe y el computador lo dé por bueno. Con la práctica verás que puedes hacerlo.



El programa que acompaña este artículo al final de la Revista te permitirá grabar un sector en particular y averiguar si éste quedó malo o no.

Finalmente, el último método que existe, es poseer un Hardware especial que se incorpora a la diskettera llamado HAPPY o bien DUPLICATOR (según la empresa que lo haya hecho), que entre otras cosas permiten el formateo de un sector en particular.

Como en todos estos métodos el computador busca un sector malo, sin especificar qué tipo de error fue, el "pirata" puede apagar su diskettera para simular el error en el momento preciso y luego volver a prenderla para que continúe la carga normalmente. Para evitar esto, se puede hacer que el programa justo a continuación de la lectura del sector malo lea un sector bueno y si no se dan las dos condiciones simultáneamente, uno malo y uno bueno, el programa no funciona. Esto va a exigir demasiada precisión en el apagado y encendido de la diskettera por parte del pirata, haciendo su misión prácticamente imposible.

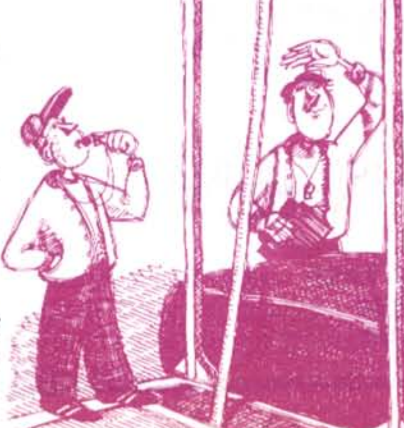
En el próximo número veremos cómo esconder, dentro del programa, las rutinas de chequeo de los sectores malos, para que no puedan ser eliminadas.

Mapa de memoria

En esta sección de tu Revista Turbo News, continuaremos explicándoles a nuestros lectores la disposición de las distintas zonas de la memoria que definen al Sistema Operativo del Atari.

736-737 $2E0-$2E1 RUNAD: En nuestro artículo de Files Binarios, hemos explicado el funcionamiento de la carga de estos archivos. En nuestra Revista Nro. 9 hicimos una completa descripción de este tipo de carga y hemos visto cómo se utilizan las posiciones de memoria $2E0 y $2E1. Cuando un archivo tipo File es terminado de cargar en la memoria del computador, estas dos posiciones de memoria son consultadas por el DOS. Si estas posiciones apuntan a una determinada zona de la memoria, el S.O. interpreta que es en esta zona donde comienza la ejecución del File.

738-739 $2E2-$2E3 INITAD: También en este artículo hemos estudiado el funcionamiento de estas direcciones de la memoria del computador. Pero para refrescar estos conceptos podemos decir que cuando se carga un File con sus respectivos Stage, si las direcciones que definen el INITAD, es decir, 738 y 739, son distintas de $00 $00, se ejecuta un llamado a la subrutina que es apuntada por estas dos direcciones para luego seguir la carga del File.



740 $2E4 RAMSIZ: Esta posición de memoria contiene la cantidad de páginas de memoria, es decir, grupos de 256 bytes que contiene nuestro computador disponibles en RAM.

741-742 $2E5-$2E6 MEMTOP: Estas dos posiciones de memoria vienen a apuntar a la última posición disponible en RAM para programas o datos del usuario. Este valor es



OFERTAS DE ANIVERSARIO

QSI
$ 2.248

QS 129 F
$ 5.945

QS II TURBO
$ 5.220

CASA MUSA CELEBRA SUS 60 AÑOS CON LOS MEJORES PRECIOS EN SOTWARE PARA ATARI

CASA MATRIZ • SAN PABLO 1055 - FONOS: 6964285 - 714223
SUCURSALES • EST. U. DE CHILE DEL METRO - LOC. 17 FONO: 724790
• PROVIDENCIA 2099 - FONO: 2319883
• APOQUINDO 6029 - LOC. 20 - FONO: 2127179
• BARROS ARANA 435 - 439 - FONO: 224907 CONCEPCION
• VICUÑA MACKENNA 820

Mapa de memoria

definido por el Sistema Operativo cuando el computador es encendido y se redefine cada vez que se presiona la tecla Reset o se modifican los modos gráficos de la pantalla. Esto es así, porque el Display List del Antic se posiciona en los siguientes bytes apuntados por MEMTOP.

743-744 $2E7-$2E8
MEMLO: Estas dos posiciones de memoria definen la primera posición disponible de la Ram para el usuario.

752 $2F0 CRSINH: Esta posición de la memoria es utilizada para hacer aparecer o desaparecer al cursor de la pantalla. Si queremos hacer desaparecer al cursor, es necesario pokear a esta posición con un 1 y ejecutar un movimiento del cursor en cualquier sentido. Esto lo podemos realizar de la siguiente manera:

```
10 POKE 752,1
20 PRINT " "
```

756 $2F4 CHBAS: Cuando en nuestra Revista Turbo news explicamos los fundamentos del Set de Caracteres en nuestro ATARI, describimos el funcionamiento de esta posición de memoria. Básicamente CHBAS viene a definirle al Sistema Operativo del Computador de dónde debe sacar la definición de los caracteres para mostrarlos en la pantalla.

764 $2FC CH: Esta posición de la memoria permanentemente contiene el código interno de la última tecla presionada. Si queremos investigar esta posición de la memoria podemos ingresar en nuestro computador el siguiente programa BASIC:

```
10 DATA=PEEK(764)
20 PRINT "TECLA PRESIONADA=
";DATA
30 POKE 764,255
40 FOR I=1 TO 250
50 NEXT I
60 GOTO 10
```

De esta manera, presionando cada tecla, podemos obtener una tabla de los valores internos de las teclas que puede ser muy útil en nuestra programación. Es muy importante que para volver a leer esta posición la pokees con un 255 o un $FF para limpiar su contenido.

Luego de la definición de estas zonas de memoria continúa en el computador la página 3, que abarca las direcciones comprendidas entre $300 y $3FF. Como la mayoría de estas direcciones han sido estudiadas en profundidad por los artículos del curso de Assembler en la Edición anterior y en la presente, sólo analizaremos las siguientes direcciones:

960-999 $3C0-$3E7
PRNBUF: Estas posiciones de la memoria del computador definen una zona especial para la comunicación con la impresora. Cada byte que se desee imprimir debe pokearse en estas posiciones, pero si utilizamos el comando del BASIC LPRINT esta operatoria es automática.

1021-1151 $3FD-$47F
CASBUF: Estas posiciones de la memoria son utilizadas por el S.O. como un Buffer para la grabación o lectura de bloques con la cassettera.

ASSEMBLER

LECCION 11

En la edición anterior de nuestra Revista Turbo news, desarrollamos gran parte del funcionamiento de la comunicación entre la Computadora y sus Periféricos.

En esta edición agotaremos estos conceptos introduciéndonos de lleno en el desarrollo práctico en lenguaje Assembler.

Para realizar esta tarea nos referiremos al programa SAVE.MAC que se encuentra listado en esta Revista en la Sección de Programas.

El objetivo de este programa es la grabación o lectura de una tabla de bytes tanto desde la cassettera o la diskettera.

Dediquemos primero nuestra atención a la grabación en la Unidad de Cinta o Cassettera.

Para grabar información en la cassettera, debemos utilizar un IOCB. Como vimos en la edición anterior de nuestra Revista, los IOCB se encuentran en la memoria en las direcciones:

IOCB0 $340
IOCB1 $350
IOCB2 $360
IOCB3 $370
IOCB4 $380
IOCB5 $390
IOCB6 $3A0
IOCB7 $3B0

Para realizar la grabación, una vez seleccionado el IOCB, debemos asegurarnos que éste



ASSEMBLER

se encuentre disponible.

Para esto lo que podemos realizar es un comando Close que cierra el canal o IOCB si es que éste se encuentra abierto.

Para cerrar el IOCB en Assembler debemos ingresar el siguiente código:

```
LDX #$10
LDA #12
STA $342,X
JSR $E456
```

En la página 22 de la Revista Turbo N° 10, desarrollamos una tabla con los valores decimales de cada comando. En el caso del Close, el comando es el número 12.

Con la primera instrucción LDX #$10, estamos seleccionando el IOCB N° 1, pues al hacer el STA $342,X en realidad nos estamos refiriendo a $352.

Al pokear el 12 en $352, en realidad le estamos definiendo al CIO que el comando que queremos desarrollar es un Close del canal o Iocb N° 1. Luego ejecutamos el llamado al Cio con la instrucción JSR $E456.

Una vez asegurado el canal, debemos abrirlo con los parámetros que lo definan. Para esto podemos ingresar el siguiente código de programa:

```
OPENCAS
     LDX #$10
     LDA #<CASSETE
     STA $344,X
     LDA #>CASSETTES
     STA $345,X
     LDA #$03
     STA $342,X
     LDA #$08
     STA $34A,X
     LDA #$80
     STA $34B,X
     JSR $E456
     RTS
CASSETTES
     .BYTE "C:",$9B
```

En este caso, la primera instrucción también define que se va a utilizar el canal N° 1. Luego en $344,X y $345,X le



indicamos al CIO qué periférico va a utilizar. El CIO, se fija a qué dirección apuntan estas dos posiciones de memoria.

En este caso, apuntan a CASSETTES, que contienen una C un : y un Return. Así se define el nombre del periférico de una forma muy similar a la utilizada por el Basic.

Luego en $342,X se define el comando que en este caso es un OPEN, es decir, un 3 como lo indica la tabla de referencia de comandos ya mencionada.

Además es necesario definir ICAX1 e ICAX2, es decir, $35A,X y $35B,X. ICAX1 viene a indicar en este caso si el canal se va a ocupar para grabación o para lectura. Si lo pokeamos con un $08, el canal será para grabación y si lo hacemos con $04 será para lectura.

ICAX2 en la cassettera viene a definir el tipo de grabación que se va a realizar. Existen dos formas de grabar un archivo en cassette y se diferencian en el hecho que al grabar cada bloque de información el motor de la cassettera se detenga o no. Si en ICAX2 pokeamos un $80 la cassettera no se detiene entre bloque y bloque.

Una vez definidos estos parámetros, podemos ejecutar la llamada al CIO para que ejecute el Open del Cassette.

Cuando el Cio devuelve el control al programa principal, luego de haber realizado el Open, el flag Negativo informa si la operación fue o no correcta. Si éste es cero el Open fue exitoso y en caso contrario existió algún error en la Unidad de Cinta. Por este motivo luego de realizar el JSR $E456 podemos consultar con un BMI si es que hubo algún error. Para ésto, el código será:

```
    JSR $E456
    BMI ERROR
    JMP OPERACIONOK
ERROR
    IMPRIMO EL CODIGO DEL
    ERROR
    Y CIERRO EL ARCHIVO
OPERACIONOK
    GRABO EN LA CINTA.
```

Asumiendo que en la apertura del archivo no existió ningún error, podemos grabar la información que necesitemos almacenar. Para esto debemos definirle al Cio los siguientes parámetros:

```
LDX #$10
LDA #11
STA $342,X
LDA #<BUFFER
STA $344,X
LDA #>BUFFER
STA $345,X
LDA #<CANTIDAD
STA $348,X
LDA #>CANTIDAD
STA $349,X
JSR $E456
BMI ERROR
```

En esta operación de grabación, existen 3 conceptos importantes:

1) $342,X se define en 11 que es el comando correspondiente a un Put Binary Record, es decir, una grabación de un bloque de bytes.

2) $344,X y $345,X apuntan a BUFFER que es la dirección de la cual el CIO va a sacar los bytes a grabar.



3) $348,X y $349,X indican la cantidad de bytes a grabar en la operación en curso.

Si la operación fue exitosa nos queda una tarea que es bien necesaria: Cerrar el IOCB o canal. Para esto nuevamente realizamos:

```
LDX #$10
LDA #12
STA $342,X
JSR $E456
```

Cuando nos referimos a la grabación de bytes en la Diskettera, el trabajo es similar. Lo único que debemos tomar en cuenta es que el nombre del dispositivo, en vez de ser "C:",$9B, debe referirse a un archivo en disco como ser:

```
"D1:FILE.BAS",$9B.
```

Las rutinas del Cio funcionan de igual manera con los distintos comandos para la Diskettera pero en este caso es necesario que un D.O.S. se encuentre cargado en la memoria del computador.

En el caso de la lectura de información desde los periféricos hacia la Unidad Central del Computador, la operatoria es similar al caso de la escritura, pero algunos de los parámetros del CIO deben cambiar.

Para asegurarnos que el canal o IOCB no está siendo ocupado, ejecutamos primero un comando ÇLOSE mediante el llamado a la rutina del CIO o $E456.

```
CLOSE
     LDX #$10
     LDA #12
     STA $342,X
     JSR $E456
```

Luego es necesario abrir el archivo. Nuevamente asumiremos el uso de la cassettera para la lectura de datos. Por lo tanto abriremos con un comando Open la cassettera o "C:".

```
OPENCAS
     LDX #$10
     LDA #<CASSETTE
     STA $344,X
     LDA #>CASSETTES
     STA $345,X
     LDA #$03
     STA $342,X
     LDA #$04
     STA $34A,X
     LDA #$80
     STA $34B,X
     JSR $E456
     RTS
CASSETTES
     .BYTE "C:",$9B
```

En la apertura del archivo, lo único que tendremos que cambiar es el valor asociado a $34A,X que en la lectura debe contener un $04.

También la rutina que se dedica a realizar la transferencia de datos desde la cassettera al computador es similar a la grabación. El único parámetro que debemos modificar es el código del comando. A $342,X le debemos pokear un $07, el cual en nuestra tabla de comandos viene a indicar un Get Binary Record o lectura de un bloque de bytes.

```
LDX #$10
LDA #07
STA $342,X
LDA #<BUFFER
STA $344,X
LDA #>BUFFER
STA $345,X
LDA #<CANTIDAD
STA $348,X
LDA #>CANTIDAD
STA $349,X
JSR $E456
BMI ERROR
```

Como podrás ver, tanto en la grabación como en la lectura, las rutinas son prácticamente iguales. Lo único que debemos modificar son los parámetros a ocupar. Por este motivo esta estructura nos permite la generación de Subrutinas que definan las distintas tareas.

Por ejemplo, podríamos definir la Rutina Close, que recibe como parámetro el número del IOCB a cerrar. También podemos definir una rutina de Open que lo ejecute dependiendo de los parámetros que reciba y por último una rutina de transferencia de bytes.

Así es como hemos definido las rutinas en el listado del programa SAVE.MAC en la sección de programas. Espero que te detengas a analizar este programa y saques de él las rutinas más importantes.

En los siguientes números de tu Revista Turbo news, desarrollaremos en este curso las Rutinas del Floating Point que son indispensables para la manipulación de números y operaciones matemáticas más complejas que la suma y la división, así que deberás estar listo para que analicemos a fondo otro de los conceptos importantes de la programación en Assembler Atari.


PROMOCION
turbo

Gráficos por computadora

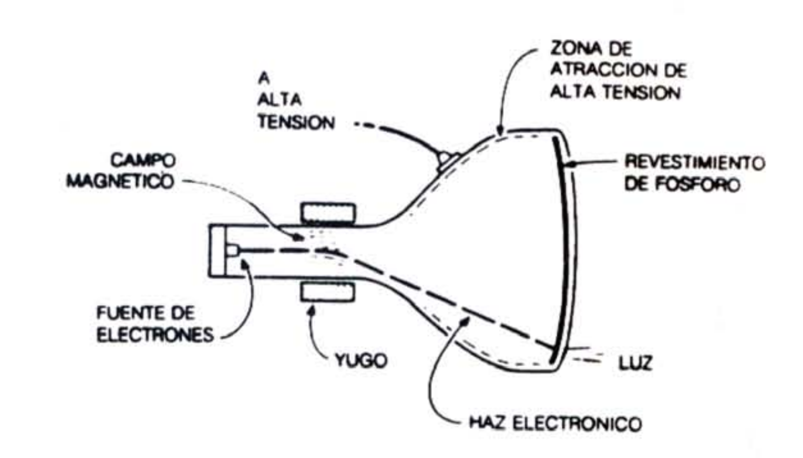
LECCION 6

Para poder entender bien cómo se hacen los Playfields, debemos estudiar primero cómo funciona el computador internamente.

La "pizarra" que utiliza en la mayoría de los casos el computador para realizar sus gráficos, es el televisor o monitor. Decimos la mayoría de los casos porque existen aparatos como impresoras y plotters que permiten realizar gráficos en papel.

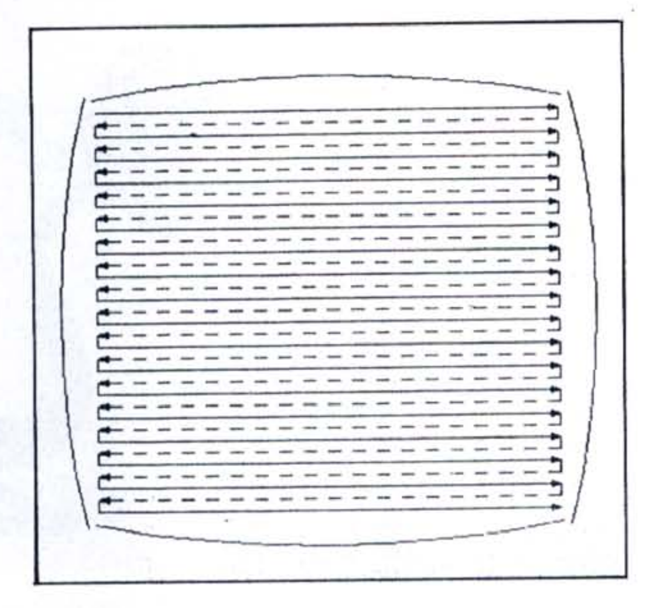
El televisor o monitor está compuesto por un tubo de rayos catódicos. Este es un tubo de vidrio con un extremo plano y un cuello largo (como se ve en el gráfico), puesto al vacío. En el extremo del cuello hay un dispositivo que emite electrones, los cuales son "disparados" hacia la parte plana. Esta se encuentra revestida de materiales especiales (fósforos), que emiten luz cuando inciden sobre ellos los electrones.

Creando un campo magnético alrededor del cuello del tubo se puede dirigir el haz de electrones al lugar de la pantalla requerido.



En el número anterior concluimos con los gráficos Player Missile. A partir de ahora verémos los escenarios o Playfields a través de los cuales dichos gráficos se mueven.

El trazo se mueve de la siguiente manera. Comienza en la esquina superior izquierda, recorre toda una fila induciendo los puntos correspondientes a la imagen que se desea formar. Luego se apaga y va al extremo izquierdo de la fila siguiente. Cuando llega al extremo derecho de la última fila, vuelve al punto de partida. Todo esto se realiza 60 veces por segundo.



De esta manera se creó una grilla de filas y columnas (estas últimas formadas por el ancho de los puntos de luz que son llamados Pixels). La información de si cada uno de los puntos deben estar prendidos o apagados, se almacena en la memoria RAM del computador. Cuando se utilizan televisores a color, en lugar de tener un cañón que dispara electrones, tenemos tres.

Asimismo cada pixel está dividido en tres, conteniendo cada uno de ellos un fósforo especial coloreado con los colores básicos: rojo, verde y azul (RGB, según sus iniciales en inglés). Cada uno de los cañones tiene la misión de disparar sus electrones sólo a la parte del pixel que le corresponde según el color. Los colores secundarios son obtenidos como combinaciones de estos tres. Para conocer el color que va a tener cada uno de los pixels debemos tener una serie de bits de información que nos diga cuál va a ser la intensidad de cada uno de los haces de electrones. Si tenemos ocho posibles intensidades de cada uno de



Gráficos por computadora

los tres colores, vamos a necesitar 3 bits para cada color para poder representarlos, ya que $2^3=8$ lo que hace un total de 9 bits por pixel y 512 colores(8*8*8). Así, por ejemplo, si la pantalla tiene 1024 filas por 1024 columnas y ocho intensidades por cada color, necesitaremos 9 bits por pixels (3+3+3) multiplicado por la cantidad de pixels de la pantalla (1024*1024) lo que hace un total de 9.437.184 bits o 1.179.648 bytes. Esto hace que el computador se vuelva sumamente costoso. Una primera forma de reducir la cantidad de memoria utilizada, es asignarle a cada pixel no el color, sino el puntero a una tabla de registro de colores. En el caso de Atari existen cuatro registros de colores, por lo tanto necesita sólo 2 bits por pixel para representarlo. La tabla a su vez tiene 8 bits, o sea que se le puede asignar a cada registro hasta 256 colores. Este sistema tiene la desventaja de no poder tener gran cantidad de colores simultáneos en pantalla pero con muy poca cantidad de memoria extra, se puede agrandar la cantidad de bits de la tabla y por lo tanto la cantidad de colores a utilizar. Si en lugar de tener 1 byte en la tabla tuviéramos 3 bytes, podríamos tener $2^{24}=16.777.216$ colores, ocupando tan solo 2 bits por pixel y 12 bytes de la tabla (4*24=96 bits dividido 8 ó sea 12 bytes). Otra gran ventaja es que al modificar uno de los registros se modifican automáticamente todos los pixels que apuntan a ese registro.

Para la selección de la definición y la cantidad de colores simultáneos que se pueden utilizar, el computador Atari utiliza los Modos Gráficos. Existen 16 modos gráficos, de los cuales 5 son de texto. A continuación veremos una tabla donde se encuentran todas las características de estos modos gráficos.

Note que a mayor definición y a mayor cantidad de colores, aumenta la cantidad de memoria necesaria.

El computador hace combinaciones de estos modos gráficos en forma automática. Siempre deja las últimas cuatro líneas de modo gráfico 0, para poner texto. Si quieres eliminarlo, debes sumar 16 al modo gráfico escogido. Así por ejemplo si haces:

GRAPHICS 1

Verás cómo te quedan en la base cuatro filas en modo gráfico 0. Si hacemos:

GRAPHICS 17

Dichas líneas desaparecen. El programador también puede hacer cualquier tipo de combinaciones alterando el Display List, como vimos en números anteriores.

Veamos un pequeño ejemplo de cada uno de los modos gráficos.

El modo gráfico 0, es el más conocido por todos nosotros, ya que es el que aparece al encender el computador para utilizar el Basic.

El modo gráfico 1 y 2, es utilizado para agrandar el tipo de letra y poder escribir en colores. Por ejemplo:



```
10 GR. 1:PRINT #6;"HOLA"
```

La instrucción Print debe ir acompañada del número de canal 6, por ser un modo gráfico. Lo mismo sucede al utilizar el modo gráfico 2:

```
10 GR.2:PRINT #6;"HOLA"
```

Para volver a las pantallas normales puedes oprimir la tecla RESET o bien, escribir GRAPHICS 0.

Si utilizas la instrucción GRAPHICS 1+16, verás que se borran las cuatro líneas de modo gráfico 0:



```
10 GR. 1+16:PRINT #6;"HOLA"
20 GOTO 20
```

Debemos agregar la instrucción 20, ya que en caso contrario, cuando el computador escriba la palabra READY, volverá automáticamente al modo gráfico 0. Para hacer letras de distintos colores, basta con combinar letras mayúsculas, minúsculas, en video inverso mayúsculas y en video inverso minúsculas. En estos dos modos gráficos los textos saldrán siempre en mayúscula y de distintos colores. Ejemplo:

```
GR.1:PRINT #6;"HoLa"
```

Para ver los restantes modos gráficos y sus diferencias, veamos el siguiente ejemplo:

```
10 GR. 3:COLOR 1:PLOT
1,1:DRAWTO 10,5
20 COLOR 2:PLOT 20,10
30 COLOR 3:PLOT 30,15
40 GOTO 40
```

Prueba ejecutar este mismo programa alterando el valor del modo gráfico.

Ten en cuenta que hay modos gráficos que no utilizan más que dos colores y por ello alguna de las rectas dibujadas serán invisibles. En la siguiente tabla podrás ver qué modos gráficos responden a qué instrucciones COLOR, SETCOLOR, POKE y cuáles son sus colores al encender el computador.

GRAFICO	COLORES	FILAS	COLUMNAS	BYTES	TEXTO
0	2	40	24	960	SI
1	5	20	24	480	SI
2	5	20	12	240	SI
3	4	40	24	240	NO
4	2	80	48	480	NO
5	4	80	48	960	NO
6	2	160	96	1920	NO
7	4	160	96	3840	NO
8	2	320	192	7680	NO
9	16	80	192	7680	NO
10	9	80	192	7680	NO
11	16	80	192	7680	NO
12	4	40	24	960	SI
13	4	40	12	480	SI
14	2	160	192	3840	NO
15	4	160	192	7680	NO



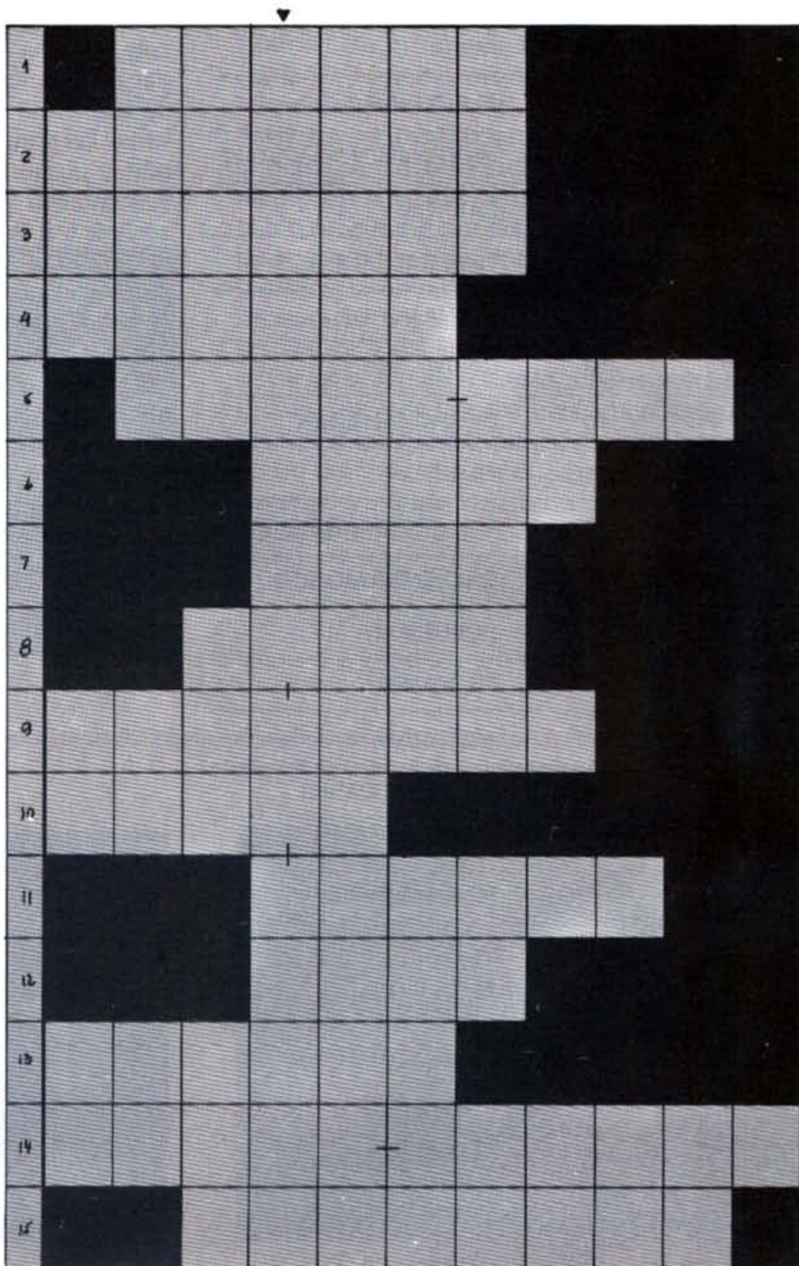
MODOS	COLORES	SETCOLOR	POKE	COLOR	DESCRIPCION
0	AZUL CLARO	1	709		CARACTERES
	AZUL	2	710		FONDO
	NEGRO	4	712		CONTORNO
1	NARANJA	0	708		CARACTER
Y	VERDE CLARO	1	709		CARACTER
2	AZUL	2	710		CARACTER
	ROJO	3	711		CARACTER
	NEGRO	4	712		CONTORNO/FONDO
3	NARANJA	0	708	1	PIXEL
5	VERDE CLARO	1	709	2	PIXEL
7	AZUL	2	710	3	PIXEL
15	NEGRO	4	712	0	PIXEL/FONDO/CONTORNO
4	NARANJA	0	708	1	PIXEL
6	NEGRO	4	712	0	PIXEL/FONDO/CONTORNO
14					
8	AZUL CLARO	1	709	1	PIXEL
	AZUL	2	710	0	PIXEL/FONDO
	NEGRO	4	712		CONTORNO
9	NEGRO	4	712	0	PIXEL/FONDO/CONTORNO
				1	PIXEL
				2	PIXEL
				3	PIXEL
	GRIS OSCURO			4	PIXEL
				5	PIXEL
				6	PIXEL
				7	PIXEL
	GRIS			8	PIXEL
				9	PIXEL
				10	PIXEL
				11	PIXEL
	GRIS CLARO			12	PIXEL
				13	PIXEL
				14	PIXEL
	BLANCO			15	PIXEL
10	NEGRO		704	0	PIXEL/FONDO/CONTORNO
	NEGRO		705	1	PIXEL
	NEGRO		706	2	PIXEL
	NEGRO		707	3	PIXEL
	NARANJA	0	708	4	PIXEL
	VERDE CLARO	1	709	5	PIXEL
	AZUL	2	710	6	PIXEL
	ROJO	3	711	7	PIXEL
	NEGRO	4	712	8	PIXEL
11	NEGRO	4	712	0	PIXEL/FONDO/CONTORNO
	NARANJA CLARO			1	PIXEL
	NARANJA			2	PIXEL
	NARANJA ROJIZO			3	PIXEL
	ROSA			4	PIXEL
	PURPURA			5	PIXEL
	AZUL PURPURA			6	PIXEL
	AZUL OSCURO			7	PIXEL
	AZUL CLARO			8	PIXEL
	AZUL CIELO			9	PIXEL
	AZUL CLARO				TURQUESA-PIXEL
	AZUL VERDOSO			11	PIXEL
	VERDE			12	PIXEL
	AMARILLO-VERDE			13	PIXEL
	NARANJA-VERDE			14	PIXEL
	NARANJA-CLARO			15	PIXEL

En el modo gráfico 9 el cambio de matiz se efectuúa con la instrucción SETCOLOR 4,n,0 o POKE 712, n.

En el modo gráfico 11 el cambio de matiz se efectúa con la instrucción SETCOLOR 4,0,n o POKE 712,n.

En el próximo número continuaremos viendo los modos gráficos.

TURBO PUZZLE:



Completa las palabras del esquema utilizando las sílabas que se dan, a razón de una letra por casilla. Una vez completo, podrás leer en la columna marcada el nombre de la técnica gráfica que sirve para esquematizar las acciones en una secuencia lógica.

1 Instrucciones escritas en un lenguaje comprensible para el computador.

2 Sistema operativo ampliado del computador.

3 Sistema numérico de base dos.

4 Traspasar al computador una información desde un diskette.

5 Tu revista de computación favorita.

6 Uno de los circuitos integrados del computador.

7 Lista de opciones entre las que hay que elegir.

8 Uno de los lenguajes del computador.

9 Entretenido juego de beisball.

10 Otro de los circuitos integrados del computador.

11 Entretenido juego con el más popular de los deportes.

12 Otro de los lenguajes del computador.

13 Tecla que sirve para cambiar la línea lógica.

Sílabas
AN BA BALL AI BO BOL CAR CO DI FUT GAR GO GO HARD JA JOY KEY LO MAS ME MON NA NIN NEWS NI NU OM PO RE RIO SIC STIC K TIC TER TUR TURN

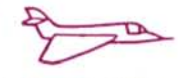
14 Famoso juego de karate.

15 Comando de palanca vertical para jugar.

RANKING DEL MES

POS. DEL MES	POS. MES ANTERIOR	TITULO
1	12	RALLY SPEEDWAY
2	8	NINJA
3	13	ELECTRICIAN
4	9	INTERNATIONAL KARATE
5	6	ZAXXON
6	15	WAR GAMES II
7	16	BOINA VERDE
8	18	ON TRACK
9	10	DESPATCH RIDER
10	21	TIME SLIP
11	2	MONTEZUMA'S REVENGE
12	1	RAID OVER MOSCOU
13	-	SUPERMAN
14	-	FROGGER I
15	-	ALLEY CAT
16	11	BOULDER DASH I
17	3	GREMLINS
18	-	SUPER COBRA
19	-	GYRUSS
20	25	GHOSTBUSTER
21	24	CRYSTAL RAIDER
22	-	POOYAN
23	5	GHOSTCHASER
24	-	SPACE INVADERS
25	17	FIGHTER PILOT II


ASCENSO


CONSTANTE

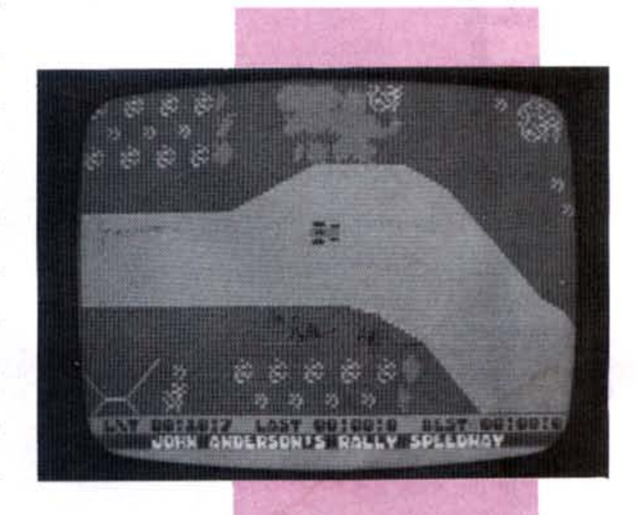

DESCENSO

Este es el Ranking correspondiente al mes de Junio, obtenido en base a las estadísticas de ventas de cassettes Turbo Software en todo Chile. Recuerda que tus preferencias también serán tenidas en cuenta, para lo cual podrás escribir a Av. Fco. Bilbao 4226, Las Condes, con los juegos de tu elección.

RALLY SPEEDWAY

En este juego, tú correrás un emocionante rally por una sinuosa carretera. El objetivo es dar una vuelta completa al circuito, tratando de realizar el menor tiempo posible. Podrás variar las condiciones de la prueba, haciendo que la pista esté húmeda, seca o con hielo, aumentar tu velocidad máxima y la aceleración de tu vehículo. Si chocas, además de recibir algunos reproches del piloto, perderás unos valiosos segundos.

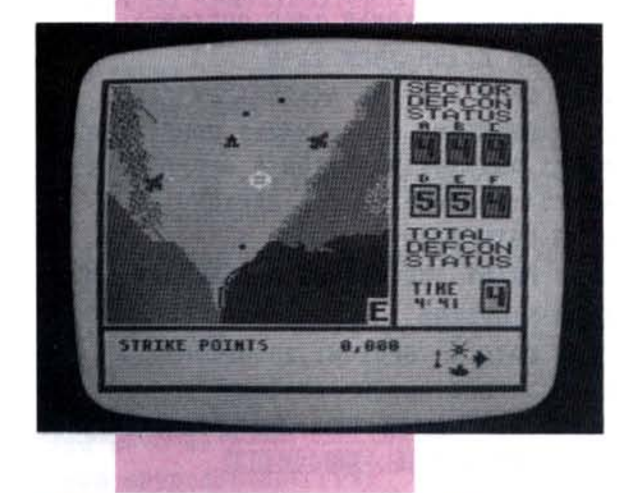
Sin duda, uno de los atractivos más interesantes de este juego, es la capacidad que tiene de poder crear nuestras propias pistas. Podrás definir curvas, rectas, cruces y agregar todos los obstáculos que desees. Es importante tener en cuenta que para crear nuestra propia pista, debes utilizar dos joysticks o alterar el único que tienes, entre los dos conectores.



SONIDO	6.5
GRAFICACION	6.7
ADICCION	6.7
PRESENTACION	6.7
PROMEDIO	6.65

WAR GAMES II

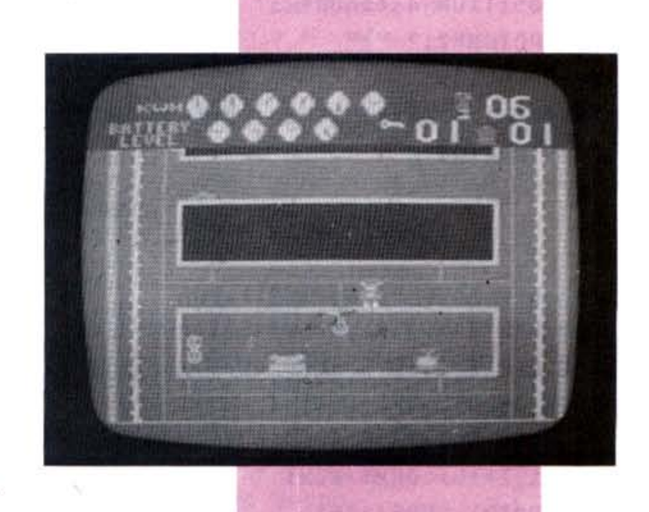
SONIDO	6.6
GRAFICACION	6.7
ADICCION	6.5
PRESENTACION	6.6
PROMEDIO	6.6



Basada en la famosa película "Juegos de Guerra". Tu misión es defender las bases y ciudades norteamericanas, de la ola de misiles termonucleares lanzados desde la Unión Soviética. El juego consta de dos pantallas. En la primera, como lo indica la foto, tienes un mapa con la visión de la situación general del combate, donde podrás determinar qué base de la ciudad se encuentra más comprometida, para acudir en su ayuda. Dicha ayuda puede llegar desde los submarinos nucleares, desde un satélite espacial defensivo, desde aviones cazas de combate o desde misiles tierra-aire. Una vez elegida la zona más amenazada, pasarás a la segunda pantalla, oprimiendo el botón del joystick, donde verás claramente al enemigo y podrás destruirlo con alguna de las armas anteriormente mencionadas.

ELECTRICIAN

En este juego, tú eres un experto electricista muy solicitado por los vecinos. El sistema eléctrico del barrio se encuentra en muy malas condiciones y cuentan contigo para repararlo.
Deberas recorrer departamento por departamento, reparando sus instalaciones. Debes apresurarte en cerrar cada uno de los circuitos, ya que mientras los estás uniendo, pueden venir los gusanos, las ratas o los murciélagos y comerse los cables (luego de conectados, de hacerlo morirán electrocutados).
Una vez arreglada una casa, deberás pasar a la siguiente por las oscuras alcantarillas de la ciudad. Estas son un verdadero laberinto de varios niveles, comunicadas por escalerillas y plagadas de alimañas y cables pelados que producen continuos cortos circuitos. Por suerte posees una linterna para observar tu alrededor.



SONIDO	6.4
GRAFICACION	6.7
ADICCION	6.6
PRESENTACION	6.6
PROMEDIO	6.575



BASIC I

```
1 REM SAVE "D:BASIC11.PRG
10 PRINT CHR$(125)
20 POKE 752,1:POKE 710,1:POSITION 15,1:PRINT "MENU PRINCIPAL"
30 POSITION 6,6:PRINT "1- OPCION1"
40 POSITION 6,8:PRINT "2- OPCION2"
50 POSITION 6,10:PRINT "3- OPCION3"
60 POSITION 6,12:PRINT "4- OPCION4"
70 POSITION 6,14:PRINT "5- OPCION5"
75 POSITION 10,17:PRINT "UTILICE CTRL ξ↑ Y CTRL ξ↓"
80 POSITION 4,6:PRINT ">":OPCION=1
90 OPEN #1,4,0,"K:"
100 GET #1,A
110 IF A=28 THEN AUX=OPCION:OPCION=OPCION-1:IF OPCION<1 THEN OPCION=5
115 IF A=29 THEN AUX=OPCION:OPCION=OPCION+1:IF OPCION>5 THEN OPCION=1
120 IF A=155 THEN 200
130 IF A<>28 AND A<>29 THEN 100
140 POSITION 4,4+AUX*2:? " ":POSITION 4,4+OPCION*2:? ">"
150 GOTO 100
200 PRINT CHR$(125);"UD. ELIGIO LA OPCION ";OPCION
```

BASIC II

```
2000 REM SAVE "D:BASIC211.PRG
2010 DIM NOM$(50),NOM2$(50),AUX$(10)
2020 FOR I=1 TO 5:READ AUX$:NOM$((I-1)*10+1,I*10)=AUX$:NEXT I
2030 FOR I=1 TO 5:READ AUX$:NOM2$((I-1)*10+1,I*10)=AUX$:NEXT I
2040 PRINT CHR$(125)
2050 POKE 752,1:POKE 710,1:POSITION 15,1:PRINT "MENU PRINCIPAL"
2060 FOR I=1 TO 5
2070 POSITION 6,4+I*2:PRINT NOM$((I-1)*10+1,I*10)
2080 NEXT I
2090 POSITION 10,17:PRINT "UTILICE CTRL ";CHR$(27);CHR$(28);" Y ";CHR$(27);CHR$(29)
2100 POSITION 6,6:PRINT "1- OPCION1":OPCION=1
2110 OPEN #1,4,0,"K:"
2120 GET #1,A
2130 IF A=28 THEN AUX=OPCION:OPCION=OPCION-1:IF OPCION<1 THEN OPCION=5
2140 IF A=29 THEN AUX=OPCION:OPCION=OPCION+1:IF OPCION>5 THEN OPCION=1
2150 IF A=155 THEN 2190
2160 IF A<>28 AND A<>29 THEN 2120
2170 POSITION 6,4+AUX*2:? NOM$((AUX-1)*10+1,AUX*10):POSITION 6,4+OPCION*2:? NOM2$((OPCION-1)*10+1,OPCION*10)
2180 GOTO 2120
2190 PRINT CHR$(125);"UD. ELIGIO LA OPCION ";OPCION
2200 DATA 1- OPCION1,2- OPCION2,3- OPCION3,4- OPCION4,5- OPCION5
2210 DATA 1- OPCION1,2- OPCION2,3- OPCION3,4- OPCION4,5- OPCION5
```

TECNICAS DE PROTECCION

```
10 DCB=768:POKE DCB+1,1
20 DIM BUFFER$(128),PRG$(6),AUX$(2)
30 BUFFER$(1,1)=" ":BUFFER$(128,128)=" ":BUFFER$(2)=BUFFER$:FOR I=1 TO 5:READ A:PRG$(I,I)=CHR$(A):NEXT I
40 PRINT CHR$(125):POSITION 13,2:PRINT "MENU PRINCIPAL"
50 POSITION 2,7:PRINT "1) GRABAR SECTOR"
60 POSITION 2,10:PRINT "2) CHEQUEAR SECTOR"
70 POSITION 10,15:PRINT "ELIJA SU OPCION: ";
80 TRAP 80:INPUT OPCION
90 IF OPCION=1 THEN GOTO 200
100 GOTO 300
200 PRINT CHR$(125):POSITION 2,10:PRINT "NUMERO DE SECTOR: ";
210 INPUT SECTOR:POKE DCB+2,87
220 POSITION 15,15:PRINT "TRABAJANDO..
```

```
."
230 GOSUB 1000
240 POSITION 15,15:PRINT "  LISTO
 "
250 POSITION 12,17:PRINT "PRESIONE RET
URN";
260 INPUT AUX$
270 GOTO 40
300 PRINT CHR$(125):POSITION 2,10:PRIN
T "NUMERO DE SECTOR: ";
310 INPUT SECTOR:POKE DCB+2,82
320 GOSUB 1000
330 IF PEEK(DCB+3)=1 THEN POSITION 10,
13:PRINT "EL SECTOR ESTA BUENO":GOTO 3
40
335 POSITION 10,13:PRINT "EL SECTOR ES
TA MALO"
340 POSITION 12,17:PRINT "PRESIONE RET
URN";
350 INPUT AUX$
360 GOTO 40
1000 ADDR=ADR(BUFFER$)
1010 ADDRHI=INT(ADDR/256)
1020 ADDRLO=ADDR-(ADDRHI*256)
1030 POKE DCB+4,ADDRLO
1040 POKE DCB+5,ADDRHI
1050 SECTORHI=INT(SECTOR/256)
1060 SECTORLO=SECTOR-(SECTORHI*256)
1070 POKE DCB+10,SECTORLO
1080 POKE DCB+11,SECTORHI
1090 X=USR(ADR(PRG$)):RETURN
2000 DATA 104,32,83,228,96
```

ASSEMBLER

```
0100 ;SAVE#D8:SAVE.MAC
0110 ;LOAD#D8:EDITOR.MAC
0120 ;
0130 ; RUTINA DE GRABACION
0140 ; DEL LISTADO. PRIMERO SE
0150 ; ENVIAN DOS BYTES "CN" QUE
0160 ; INDICAN QUE EL PROGRAMA A
0170 ; GRABAR O LEER FUE GENERADO
0180 ; CON ESTE PROGRAMA, PARA LUEGO
0190 ; RECONOCER FORMATOS INCORRECTOS
0200 ; LUEGO SE GRABA LA CANTIDAD DE
0210 ; BYTES A TRANSFERIR EN EL FORMA
0220 ; TO LO Y HI Y A CONTINUACION EL
0230 ; VALOR DE FINTAB. UNA VEZ INDI-
0240 ; CADOS ESTOS PARAMETROS SE GRA-
0250 ; BA EL LISTADO DEL PROGRAMA.
0260 ;
0270 ; FORMATO DE GRABACION
0280 ;
0290 ;      "CN"
0300 ;      CANTIDAD DE BYTES LO
0310 ;      CANTIDAD DE BYTES HI
0320 ;      FINTAB
0330 ;      FINTAB+1
0340 ;
0350 ;      DATOS
0360 ;
0370 ;
0380 ; SI NO HAY LINEAS EN EL LISTADO
0390 ; SE VUELVE AL PROGRAMA
0400 ; PRINCIPAL
0410 ;
0420 FILESAV
0430 ;
0440      LDA FINTAB+1
0450      CMP # >TABLA
0460      BNE HAYALGO
0470      LDA FINTAB
0480      CMP # <TABLA
0490      BNE HAYALGO
0500      RTS
0510 ;
0520 HAYALGO
0530 ;
0540 ; SE VA A GRABAR DESDE LA
0550 ; POSICION ROTULOGRABO EN
0560 ; ADELANTE. ENTONCES YA LA
0570 ; DEFINO CON LOS BYTES "CN"
0580 ;
0590      LDA #'C
0600      STA ROTULOGRABO
0610      LDA #'N
0620      STA ROTULOGRABO+1
0630 ;
0640 ; IMPRIMO EN LA PANTALLA EL
0650 ; MENSAJE PARA QUE EL USUARIO
0660 ; DEFINA SI VA A UTILIZAR LA
0670 ; CASSETTERA O LA DISKETTERA.
0680 ;
0690      JSR DISPOSITIVO
0700 ;
0710 ;
0720 ; TOMO UNA TECLA DEL EDITOR
0730 ; Y LA IMPRIMO EN LA PANTALLA
0740 ; PARA DEFINIR SI VAMOS A UTI-
0750 ; LIZAR LA DISKETTERA O LA
0760 ; CASSETTERA.
0770 ;
0780      JSR $F2F8
0790      STA GUARDOA
0800      JSR $F2B0
0810 ;
0820 ESPERORET
0830 ;
0840      JSR $F2F8
0850      CMP #$9B
0860      BNE ESPERORET
0870      LDA GUARDOA
0880      CMP #'C
0890      BEQ CASSETTE
0900      CMP #'D
0910      BEQ DISKETTE
```

```
0920      JMP FINSAVE
0930 ;
0940 ; EL PERIFERICO ELEGIDO POR EL
0950 ; USUARIO FUE LA CASSETTERA
0960 ;
0970 CASSETTE
0980 ;
0990 ; ME ASEGURO QUE EL  CANAL NRO. 1
1000 ; ESTE CERRADO PARA REALIZAR EL
1010 ; OPEN
1020 ;
1030      LDX #$10
1040      JSR CLOSE
1050      LDA #$08     ; $04=READ
1060      STA DSTATS
1070      LDA # <CASS
1080      LDY # >CASS
1090      LDX #$10
1100 ;
1110 ; EJECUTO EL OPEN Y SI NO HAY
1120 ; ERRORES GRABO LA TABLA DE
1130 ; INSTRUCCIONES EN LA CINTA.
1140 ;
1150      JSR OPEN
1160      BPL GRABE
1170      JMP ERROR
1180 GRABE
1190      JMP GRABETABLA
1200 ;
1210 ; SI EL PERIFERICO FUE LA
1220 ; DISKETTERA, CONSULTO SU
1230 ; NUMERO PUES PODEMOS TRABAJAR
1240 ; CON CUALQUIER UNIDAD DE DISCO
1250 ; INCLUSIVE CON LA D8:
1260 ;
1270 DISKETTE
1280      JSR DRIVE
1290      LDA #'1
1300      JSR $F2B0
1310      LDA #'
1320      JSR $F2B0
1330      JSR $F2F8
1340      CMP #$9B
1350      BNE RANGOK
1360      LDA #'1
1370      STA DUNIT
1380      JMP TOMONAME
1390 RANGOK
1400      CMP #'1
1410      BCS OK
1420      JMP FINSAVE
1430 OK
1440      CMP #'9
1450      BCC POKEDUNIT
1460      JMP FINSAVE
1470 POKEDUNIT
1480      CMP #'5
1490      BCC NO8
1500      LDA #'8
1510 NO8
1520      STA DUNIT
1530      JSR $F2B0
1540 ESPRET
1550      JSR $F2F8
1560      CMP #$9B
1570      BNE ESPRET
1580 ;
1590 ; LE PIDO AL USUARIO EL NOMBRE
1600 ; DEL ARCHIVO A UTILIZAR PARA
1610 ; HACER EL OPEN
1620 ;
1630 TOMONAME
1640      JSR NOMBRE
1650      JSR INGRESONOMBRE
1660      LDA #$08     ; $04=READ
1670      STA DSTATS
1680      LDA # <DEVICE
1690      LDY # >DEVICE
1700      LDX #$10
1710 ;
1720 ; EJECUTO EL OPEN DEL PERIFERICO
1730 ; Y SI NO EXISTE ALGUN ERROR SE
1740 ; GRABA LA TABLA.
1750 ;
1760      JSR OPEN
1770      BPL GRABETABLA
1780      JMP ERROR
1790 ;
1800 ; GRABO LA TABLA DE INSTRUC-
1810 ; CIONES EN EL PERIFERICO
1820 ; UTILIZANDO LA RUTINA CIO.
1830 ;
1840 GRABETABLA
1850      SEC
1860      LDA FINTAB
1870      STA GUARDOFINTABLO
1880      SBC # <TABLA
1890      STA CANTIDADLO
1900      LDA FINTAB+1
1910      STA GUARDOFINTABHI
1920      SBC # >TABLA
1930      STA CANTIDADHI
1940      CLC
1950      LDA CANTIDADLO
1960      ADC #6
1970      STA BYTES
1980      LDA CANTIDADHI
1990      ADC #0
2000      STA BYTES+1
2010      LDA #$0B     ; $07=READ
2020      STA DSTATS
2030      LDA # <ROTULOGRABO
2040      STA BUFFERDES
2050      LDA # >ROTULOGRABO
2060      STA BUFFERDES+1
2070      LDA BYTES
2080      LDY BYTES+1
2090      LDX #$10
2100      JSR CIOEXE
2110      BPL DOCLOSE
2120 ;
2130 ; SI SE PRODUCE ALGUN ERROR
2140 ; SE IMPRIME EN LA PANTALLA
2150 ; EL NUMERO DE ERROR UTILIZANDO
2160 ; LAS RUTINAS DEL FLOATING POINT
2170 ;
2180 ERROR
2190      STY FR0
2200      LDA #0
2210      STA FR0+1
2220      JSR $D9AA
2230      JSR $D8E6
2240      LDA # <ROTERIO
2250      STA MANDO
```

```
2260      LDA # >ROTERIO
2270      STA MANDO+1
2280      JSR IMPRIMO
2290      LDY #0
2300      STY GUARDOY
2310 LOPNNN
2320      LDA ($F3),Y
2330      BMI ULLL
2340      JSR $F2B0
2350      LDY GUARDOY
2360      INY
2370      STY GUARDOY
2380      JMP LOPNNN
2390 ULLL
2400      AND #$7F
2410      JSR $F2B0
2420 DOCLOSE
2430      LDX #$10
2440      JSR CLOSE
2450 FINSAVE
2460      RTS
2470 ROTERIO
2480      .BYTE "}ERROR EN I/O NUMERO: *"
2490 DISPOSITIVO
2500      LDA # <DEVIC
2510      STA MANDO
2520      LDA # >DEVIC
2530      STA MANDO+1
2540      JSR IMPRIMO
2550      RTS
2560 DEVIC
2570      .BYTE $9B,"DISPOSITIVO [D/C]: *"
2580 DRIVE
2590      LDA # <DRIV
2600      STA MANDO
2610      LDA # >DRIV
2620      STA MANDO+1
2630      JSR IMPRIMO
2640      RTS
2650 DRIV
2660      .BYTE $9B,"DRIVE [1-4/8]: *"
2670 NOMBRE
2680      LDA # <NAME
2690      STA MANDO
2700      LDA # >NAME
2710      STA MANDO+1
2720      JSR IMPRIMO
2730      RTS
2740 NAME
2750      .BYTE $9B,"ARCHIVO: *"
2760 ;
2770 ; INGRESO A TRAVES DEL CIO
2780 ; DESDE EL TECLADO EL NOMBRE
2790 ; DEL FILE.
2800 ;
2810 INGRESONOMBRE
2820      LDX #11
2830      LDA #32
2840 LOOPLIMPIO
2850      STA FILE,X
2860      DEX
2870      BPL LOOPLIMPIO
2880      LDX #0
2890      LDA #5
2900      STA $0342,X
2910      LDA # <FILE
2920      STA $0344,X
2930      LDA # >FILE
2940      STA $0345,X
2950      LDA #12
2960      STA $0348,X
2970      LDA #0
2980      STA $0349,X
2990      JSR $E456
3000      RTS
3010 CLOSE
3020      LDA #$0C
3030      STA $0342,X
3040      JSR $E456
3050      RTS
3060 OPEN
3070      STA $0344,X
3080      TYA
3090      STA $0345,X
3100      LDA #$03
3110      STA $0342,X
3120      LDA DSTATS
3130      STA $034A,X
3140      LDA #$80
3150      STA $034B,X
3160      JSR $E456
3170      RTS
3180 CIDEXE
3190      STA $0348,X
3200      TYA
3210      STA $0349,X
3220      LDA DSTATS
3230      STA $0342,X
3240      LDA BUFFERDES
3250      STA $0344,X
3260      LDA BUFFERDES+1
3270      STA $0345,X
3280      JSR $E456
3290      RTS
3300 BUFFERDES
3310      .WORD 0
3320 DEVICE
3330      .BYTE "D"
3340 DUNIT
3350      .BYTE "1"
3360      .BYTE ":"
3370 FILE
3380      .BYTE "LISTADOS.NUM",$9B
3390 CASS
3400      .BYTE "C:",$9B
3410 BYTES
3420      .WORD 0
3430 DSTATS
3440      .BYTE 0
3450 FLAGPUNTO
3460      .BYTE 0
3470 ;
3480 ; EJECUTO EL MISMO TRABAJO
3490 ; QUE EN LA GRABACION PERO
3500 ; EN ESTE CASO CON LOS PARA-
3510 ; METROS QUE LE INDIQUE AL
3520 ; CIO QUE LA INTENCION ES
3530 ; CARGAR UN PROGRAMA DESDE
3540 ; EL PERIFERICO.
```

```
3550 ;
3560 LOAD
3570        JSR DISPOSITIVO
3580        JSR $F2F8
3590        STA GUARDOA
3600        JSR $F2B0
3610 WRET
3620        JSR $F2F8
3630        CMP #$9B
3640        BNE WRET
3650        LDA GUARDOA
3660        CMP #'C
3670        BEQ ESCAS
3680        CMP #'D
3690        BEQ ESDISK
3700        JMP FINLOAD
3710 ESCAS
3720        LDX #$10
3730        JSR CLOSE
3740        LDA #$04
3750        STA DSTATS
3760        LDA # <CASS
3770        LDY # >CASS
3780        LDX #$10
3790        JSR OPEN
3800        BPL LEA
3810        JMP ERROR
3820 LEA
3830        JMP LEATABLA
3840 ESDISK
3850        JSR DRIVE
3860        LDA #'1
3870        JSR $F2B0
3880        LDA #'
3890        JSR $F2B0
3900        JSR $F2F8
3910        CMP #$9B
3920        BNE RANGOK?
3930        LDA #'1
3940        STA DUNIT
3950        JMP TOMONAME1
3960 RANGOK?
3970        CMP #'1
3980        BCS OK1
3990        JMP FINLOAD
4000 OK1
4010        CMP #'9
4020        BCC POKEDUNIT1
4030        JMP FINLOAD
4040 POKEDUNIT1
4050        CMP #'5
4060        BCC NO81
4070        LDA #'8
4080 NO81
4090        STA DUNIT
4100        JSR $F2B0
4110 TAKERET
4120        JSR $F2F8
4130        CMP #$9B
4140        BNE TAKERET
4150 TOMONAME1
4160        JSR NOMBRE
4170        JSR INGRESONOMBRE
4180        LDA #$04
4190        STA DSTATS
4200        LDA # <DEVICE
4210        LDY # >DEVICE
4220        LDX #$10
4230        JSR OPEN
4240        BPL LEATABLA
4250        JMP ERROR
4260 LEATABLA
4270        LDA #$07
4280        STA DSTATS
4290        LDA # <ROTULOGRABO
4300        STA BUFFERDES
4310        LDA # >ROTULOGRABO
4320        STA BUFFERDES+1
4330        LDA #$06
4340        LDY #0
4350        LDX #$10
4360        JSR CIOEXE
4370        BPL DIRECOK
4380        JMP ERROR
4390 DIRECOK
4400        LDA ROTULOGRABO
4410        CMP #'C
4420        BEQ PRIMEROK
4430        JMP BADFILE
4440 PRIMEROK
4450        LDA ROTULOGRABO+1
4460        CMP #'N
4470        BEQ OKFILE
4480        JMP BADFILE
4490 OKFILE
4500        LDA GUARDOFINTABLO
4510        STA FINTAB
4520        LDA GUARDOFINTABHI
4530        STA FINTAB+1
4540        LDA #$07
4550        STA DSTATS
4560        LDA # <TABLA
4570        STA BUFFERDES
4580        LDA # >TABLA
4590        STA BUFFERDES+1
4600        LDA CANTIDADLO
4610        LDY CANTIDADHI
4620        LDX #$10
4630        JSR CIOEXE
4640        BPL DOCLOSE1
4650        JMP ERROR
4660 DOCLOSE1
4670        LDX #$10
4680        JSR CLOSE
4690 FINLOAD
4700        RTS
4710 BADFILE
4720        LDA # <ROTBADFILE
4730        STA MANDO
4740        LDA # >ROTBADFILE
4750        STA MANDO+1
4760        JSR IMPRIMO
4770        RTS
4780 ROTBADFILE
4790        .BYTE $9B,"}ERROR DE FORMATO"
4800        .BYTE $9B,"*"
```


Turbo Software
LINEA LIGHT GUN PARA ATARI
Turbo Software
LINEA EDUCACIONAL PARA ATARI
Turbo Software
ATARI
La nueva generación de Software para Computadores Atari

QuickShot®
by Bondwell

REPRESENTANTE EXCLUSIVO PARA CHILE
LUIS A. VERDUGO AGUILERA
MONJITAS: 843 OF. 911 - C
TELEFONOS:339219 - 6321966
FAX: 02-395977
TELEX: 242283 VCF CL
SANTIAGO - CHILE



Atari y Commodore. Sega, Atari, Commodore, Amstrad y MSX
IBM PC/XT/AT IBM PC/XT/AT y Apple IIe/c
Sega, Sears, Atari, Commodore, MSX y Nintendo Entertainment System

DISTRIBUIDORES:
SANTIAGO: A. PARIS • FALABELLA • CASA ROYAL • SUPERTIENDAS ABC • DISTRIBUIDORA DELANO • RIPLEY • HITES • IMACO • ESTADO OFERTA • TELEAUDIO • COMPUMANQUE • COMPUCENTER • MICROCENTRO • INFOLAND-E. YAÑEZ • **VIÑA DEL MAR:** INSIS • SELECTRONIC • SUPERTIENDAS ABC • FALABELLA • RIPLEY • **VALPARAISO:** SELECTRONIC • COMPUTRONIC • SUPERTIENDAS ABC • **TEMUCO:** FALABELLA • SUPERTIENDAS ABC • **CONCEPCION:** FALABELLA • SUPERTIENDAS ABC • RIPLEY • SERV. COMPUTACIONAL CONCEPCION LTDA. • CRECIC S.A. • **VALDIVIA:** DOMBURGO.